AI enabled Automation in Software Development: Tapping the power of Large Language Models

Sohail Iqbal
School of Electrical Engineering & Computer Science
National University of Sciences and Technology (NUST),
Islamabad, Pakistan
sohail.iqbal@seecs.nust.edu.pk

Khuram Shahzad
School of Electrical Engineering & Computer Science
National University of Sciences and Technology (NUST),
Islamabad, Pakistan
kshahzad.dphd18seecs@seecs.edu.pk

Muhammad Moazam Fraz
School of Electrical Engineering & Computer Science
National University of Sciences and Technology (NUST),
Islamabad, Pakistan
moazam.fraz@seecs.edu.pk

Abstract—**Around the globe, 26.9 million software developers are working tirelessly to meet our software needs of the day. With the raise of the Metaverse and the gaming industry, the requirements and needs of software engineers will become too high to meet. Therefore, it becomes highly important to automate the entire software development process, by tapping the power of the recently developed ground-breaking AI tools. In this paper, we handle above mentioned problem by proposing a novel framework for AI-based automated software development. Our proposed framework utilizes recent technological advancements in AI tools to make it easier to build software quickly. Our modular approach simplifies and expedites several phases of the software development process by building on the shortcomings of the current frameworks. To generate revenue, our suggested framework offers a number of business models from its adoption. We expect that our idea will inspire others to make unique contributions to the development of automated software in order to stimulate the economy of our society.**

Index Terms—**software development, automation, natural language, artificial intelligence, large language models, GNU compiler collection.**

I. INTRODUCTION

Nowadays, software can be found just about everywhere. High-quality software is essential to the smooth running of almost every facet of modern life, from tele-medicine and tele-marketing to film industry and gaming industry [1]. Organizations may improve their efficiency and productivity with the aid of specialised tools and apps built via software development. Its significance also stems from the fact that it paves the way for the development and dissemination of novel software products, the likes of which may be used to address pressing societal issues, provide entertainment, and teach people all over the globe. There are a number of potential challenges that can occur during software development, including timely delivery, complexity, changing requirements, limited resources, quality assurance, and maintenance [2]. Furthermore, software development is expensive because it requires software engineers to overcome the inherent complexity of software, while avoiding errors and yet delivering fully functioning software products on schedule [3]. Therefore, there is always a need for new developments in software tools and processes that can improve the dependability and maintainability of a software. Researchers are actively working on the new approaches to create better software while also reducing its complexity.

Automated software development refers to the practise of using tools and procedures that speed up and simplify different stages of the process of developing software. This might consist of a variety of things, such as automated testing, code generation, and deployment, among other things [2]–[4]. It offers a number of benefits, as follows:

1) **Increased efficiency:** Automated methods, which may be quicker and more accurate than human processes, can assist to minimize the amount of time and effort that is necessary to build software.
2) **Improved quality:** Automated testing, as opposed to manual testing, may be used to identify problems and flaws in software in a more timely and accurate manner. This has the potential to contribute to an overall improvement in the quality of the program.
3) **Reduced costs:** As a result of its ability to eliminate the need for human work force, automated software development may contribute to a reduction in the human cost that are involved with software development.
4) **Increased consistency:** Automated processes may assist to guarantee that software is produced in a consistent way, which can be vital for preserving the quality and integrity of the product over time. This can be accomplished by ensuring that the software is developed in a manner that is adhering to the accepted standards of implementation.
5) **Improved economy:** Software development has had a substantial effect on the economy, both as a contributor

to economic expansion and as a source of employment. With automated software development practices the delivery time of a software is significantly reduced, which improves the overall economics of a society.

The study of artificial intelligence (AI) is expanding rapidly, making inroads into a wide variety of industries and research fields. Sub-fields of artificial intelligence that focus on data processing and modelling include machine learning, natural language processing (NLP) and deep learning. With the recent technological advancements, AI is being widely used in automated software development to automate a number of operations, including code generation, testing, and deployment. It is also being applied to generate new software tools and technologies that may increase the efficiency and productivity of the software development process [5]. A wide variety of resources exist for developing AI-powered software. Examples of such resources include, machine learning frameworks (e.g., TensorFlow, PyTorch, and scikit-learn), NLP libraries (e.g., NLTK and spaCy), AI platforms (e.g., Google Cloud AI Platform and Amazon SageMaker), and robotics frameworks (ROS and MoveIt). However, the recent emergence of AI-powered automated code generation tools (e.g., GitHub Co-Pilot, Tabine and CodeT5, to name a few) has made a substantial contribution to conceptualizing methodologies for automating software development. GitHub Co-Pilot, supported by millions of public code repositories, outperforms in reliability, versatility, and performance when compared to other AI-powered automated code generation tools [6].

A software development framework play a key role for the programmers to improve productivity, reduce cost, enhance code quality, improve security, timely delivery, and to make it easier to build and maintain software over time. In this paper, we propose a novel framework for AI-based automated software development. There are a number of issues with the currently available frameworks for this problem, including a lack of scalability, modularity, and a sustainable business model. In addition, the majority of them are not universal and instead function well for a particular application only.

Organization: The rest of the article is structured as follows: A short overview of the related literature is presented in Section II. In Section III, we formally define our problem statement. In Section IV, we present the system model of our proposed framework for automated software development. The business model of our proposed framework is presented in Section V, and we discuss the results and novelties of our work in Section VI. Finally, in Section VII, we conclude this article.

II. RELATED WORK

In 2020, A. Dubey et al. proposed a novel framework for encouraging the concept of human and AI collaboration [7]. Although the proposed framework reduces the overall development effort, it is not scalable and does not follow a modular approach for implementation. Furthermore, the framework does not support any effective business model. In the same year, S. Jüngling et al. proposed an approach to incorporate AI-based solutions in the development life cycle of a software [8]. Although the proposed approach facilitates in saving cost of software development, it does offer any effective business model and is not scalable. Moreover, the framework is not universal and only focuses on the recognition of anonymous passenger.

In 2021, C. Kaymakci et al. proposed a holistic framework to exploit AI tools in the industrial applications [9]. Although the proposed framework promotes modular implementation, it is does not support any effective business model. Besides, it is not generalized and focuses only on the manufacturing system. In the same year, P. William et al. exploited agile software development framework for task allocation using machine learning approach [10]. Although the proposed approach improves the efficiency of software development process and is universal, it does not make use of the automated code generation tools to reduce cost. Moreover, the framework does not have any business model attached.

In 2022, H. Wu et al. proposed a framework exploiting graph-based neural network for an automated software development [11]. Although, the proposed framework does reduce software development cost by automatically triaging the bugs report and is scalable, it is not generalized and does not have any business model for contributing to the economy. Moreover, the proposed framework does not make use of any AI-based code generation tools. In the same year, A. Azzazi et al. proposed a framework by making use of the knowledge-based system to the performance of software development [12]. Although the proposed framework reduces cost and enhances security of the software development process, it is not generalized and does not make use of any AI-based code generation tools.

III. PROBLEM FORMULATION

The problem statement is formally stated in this section in light of the findings from the previous section. We can analyze that different researcher have contributed to various challenges for designing a framework for an automated software development exploiting AI tools. Nevertheless, extensive research is still required to design an optimal framework for dealing with various factors in this context. The objective of this research is to address the following challenges in an optimal way, while designing a framework for an automated software development based on AI tools:

1) **AI-based coding:** Usage of AI techniques to automate the process of generating code.
2) **Reduce expenses:** Reduce the labour cost for developing a software.
3) **Scalability:** It allows a system to evolve and meet new requirements.
4) **Modularity:** Design of a system or framework into reusable units that may be used in many ways to serve a variety of purposes.

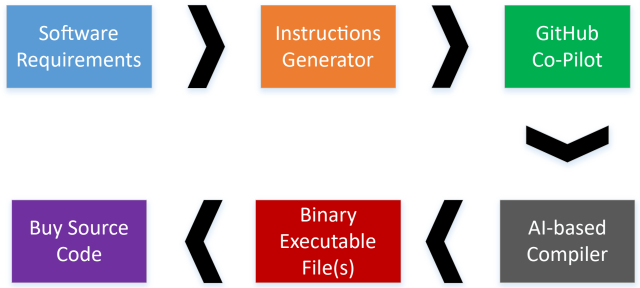

Fig. 1. High level model of our proposed framework for AI powered automated software development.

5) **Universal:** It offers a single comprehensive solution to facilitate the completion of a broad range of software projects.
6) **Business Model:** It provides a blueprint to the business organizations for generating revenue and to make profit

IV. SYSTEM MODEL

In this section, the the system model of our framework is formalized. A high level flow of various phases of our proposed framework has been depicted in Fig. 1. A detailed description of each phase has been provided one after another in the following sub-sections.

A. Software requirements

Software requirements are gathered at this phase based on business needs of a customer. When designing, developing, and deploying a software system, it is important to take into account the requirements that have been specified by a customer. These requirements are often categories into two areas, i.e., functional and non-functional software requirements. Functional requirements specify what the software should do and be able to do, whereas non-functional requirements outline limitations and expectations in areas like performance, security, and user friendliness. Requirements for software are crucial because they spell out in great detail what the software should be able to accomplish, how it should act, and how it should be used. They act as a foundation for the development of a software and guarantee that the finished software satisfies the needs and expectations of its users.

B. Instructions generator

At this phase, the software requirements, collected at the last phase, are broken down into a comprehensive set of instructions that will be passed on to the subsequent phase of the framework. It can be conceived as an AI-based tool that is used for the purpose of automatically generating instructions in accordance with the requirements of a software application. Instructions generators for software requirements that are powered by AI may save time and effort by automatically generating instructions based on the requirements and constraints of a software. This also helps to guarantee that the instructions are accurate and up-to-date. The derived instructions are then forwarded one at a time to the successive phase of the framework.

C. AI-based automatic code generator (GitHub Co-pilot)

At this phase, code is generated automatically using the AI-based tools based on the detailed instructions from the last phase of the framework. It is important to note that the code is generated against each individual instruction received from the last phase. Our framework allows to exploit any feasible AI-based code generator tool, such as GitHub Co-pilot, Chat GPT, Tabine, and CodeT5, to name a few. However, due to the versatility, reliability and performance of GitHub Co-Pilot, we recommend using it while implementing the proposed framework. Automated coding is made possible with the use of code generators, which take inputs in the form of instructions and produce code on the fly. They help developers save time and avoid repetitive tasks by generating enormous volumes of code that would otherwise have to be hand-written. The AI powered code generator tools instantly provide a very high quality of code, which contributes towards increased productivity, consistency, ease of use and cost-effectiveness.

D. AI-based compiler

At this phase, the code generated by the last phase is gathered, assembled and compiled with the help of an AI-based compiler. A compiler is a piece of software that takes source code written in one programming language and converts it into machine code. It is a crucial component of the software engineering process because it facilitates the writing of code in a high-level language that is more comprehensible by humans and then translates that code into a form that computers can execute. A compiler that makes use of AI techniques to examine and enhance source code is called an AI-based compiler. It does its job by inspecting the source code of a software with the use of machine learning algorithms to find places where improvements may be made, such as in the number of instructions used or the amount of unused code. The output of this phase contains both binary file(s) of the assembled and compiled code along with its original source code.

E. Binary executable file(s)

At this phase, the client receives the binary executable file(s) of the software for review and quality assurance purposes, after which they may determine whether or not it satisfies the requirements established at phase IV-A of the framework. It can consist of a single file or multiple files that includes compiled code that can be executed on a computer. It provides benefits like portability, efficiency, security and compatibility, to name a few. The binary files does not support editing and a human can not parse them. The inner workings of a piece of software are potentially the intellectual property of its owner, and sharing the source code for that software may disclose those inner workings to the public. Protecting this intellectual property may be aided by exchanging binary files instead of the source code.

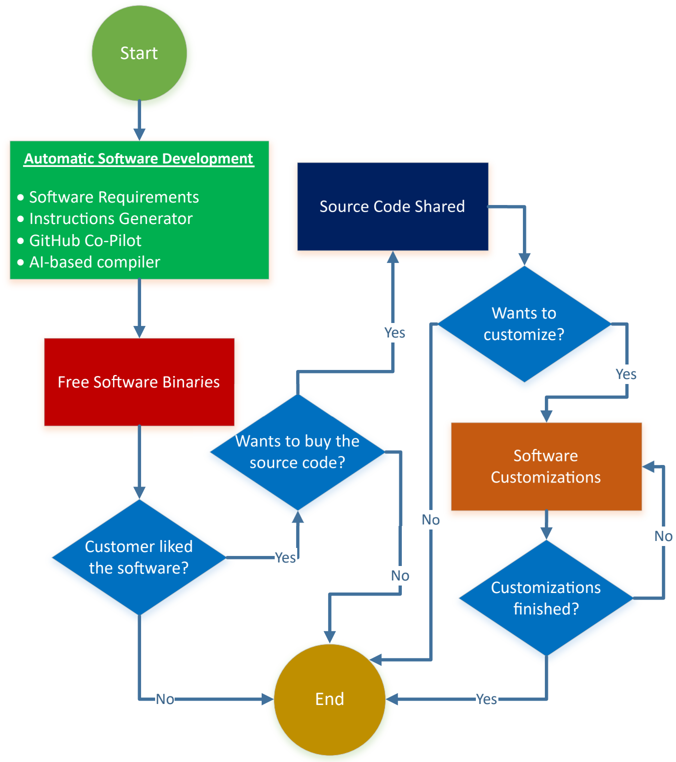

Fig. 2. High level business model of our proposed framework.

F. Buy source code

At this phase, the client can buy the source code of the software developed by the framework. In case the software, shared through binary executable file(s) in the last phase, satisfies the requirements of the client, they opt to purchase its source code. Once the source code is acquired, the buyer has full ownership of the software and is free to make any changes or use it anyway they see fit.

V. BUSINESS MODEL

A business model is a blueprint used by an organization to outline how it will generate revenue and earn a profit. It describes the items or services the company will provide, the target market it will sell to, and the operating cost it will incur. There are several types of business models, and the one that is best suitable for a given firm will based on a variety of criteria, such as the nature of the goods or services being delivered, the intended market, and the competitive environment. Some examples of typical kinds of business models include B2B (business-to-business) model, B2C (business-to-consumer) model, C2C (consumer-to-consumer) model, subscription model, freemium model and sharing economy model, to name a few [13]. Our proposed framework supports both B2B and B2C business models, as both businesses and direct consumers can opt for the development of a software that fulfil their needs.

A high level business model of our proposed framework has been depicted in Fig. 2. We can analyze that the flow starts from the automatic software development process, which coves phases like gathering software requirements, generating instructions, utilizing GitHub Co-pilot to generate automatic code, and using an AI-based compiler. This process provides free software binaries to the customer which they can test to verify whether the software meets the requirements or not. In case the customer likes the software, they can opt for buying its source code. After that, if the client wishes to use the service for software customization, the procedure continues there and concludes after the customization is complete; otherwise, it terminates without going there.

Speaking on the delivery model of this framework, it can be offered as SaaS (Software as a Service) solution to the end users [14]. To put it simply, it is a concept wherein a piece of software is hosted on the Cloud and made accessible to users through an online interface. Consumers may use a web browser to get access to the software without having to download and install it on their own computer or server. The SaaS services are paid for on a subscription basis, usually on a monthly or yearly basis. So, this can be used as a another business model to generate revenue from our proposed framework.

VI. RESULTS AND DISCUSSION

In this section, we assess the novelties offered in our research by comparing the results of our proposed framework with those of current frameworks for the development of automated software. In Table I we show how our framework is different from others in the literature. We can analyze that our proposed framework, outclass all other frameworks in the literature when comparing on the criteria defined in the Section II above. To the best of authors' knowledge, none of the existing frameworks properly demonstrated the integration of AI-based code generation for the automated development of a software. This increases the efficiency of software development by expediting the development process. AI algorithms can be trained on code that is of a high quality and free of mistakes, which can help to minimize the amount of faults and errors that are present in the output code. This also helps in improving the accuracy of the developed software. In addition, AI-powered code generators may assist to guarantee that newly generated code adheres to the format and standards of industry-standard coding practises, which can be a significant benefit.

We can also observe that by using AI tools, we are able to lower the amount of human engagement in the development process. This, in turn, results in a reduction in the cost of labour as well as the total expenditures associated with the development of software. Businesses and other organizations may save costs across their whole information technology budgets by reducing the cost of the software they need. Having reduced expenses for software allows businesses to be more competitive in their respective marketplaces since they are able to provide clients with cheaper pricing while still retaining their profitability. Small firms and organizations with limited resources may find it simpler to obtain and employ new

TABLE I

COMPARISON OF OUR PROPOSED FRAMEWORK ON AI BASED AUTOMATED SOFTWARE DEVELOPMENT WITH THE EXISTING APPROACHES

Authors	AI-based coding	Reduce expenses	Scalability	Modularity	Universal	Business Model
A. Dubey et al. [7]	Poor	Fair	×	×	×	×
S. Jungling et al. [8]	Fair	Fair	×	×	×	✓
C. Kaymakci et al. [9]	Fair	Fair	✓	✓	×	×
P. William et al. [10]	Poor	Poor	×	×	✓	×
H. Wu et al. [11]	Poor	Good	✓	✓	×	×
A. Azzazi et al. [12]	Poor	Good	×	×	✓	×
Proposed Algorithm	Good	Excellent	✓	✓	✓	✓

technology if the cost of software is reduced, which may help these organizations remain competitive in their respective marketplaces. Moreover, it is possible that businesses and organizations may become more efficient and productive if they switched to a more cost-effective software.

We can also analyze our framework from the scalability standpoint. Our framework accommodates for the future growth and expansion, allowing us to easily add new phases in its life cycle and to help evolve the system to meet new challenges. We can also examine that our framework promotes modular architecture, which may be used in a variety of ways to accommodate a broad range of functionality. A framework that is modular can be updated and maintained more easily since changes to one module do not impact the whole system. Since individual modules can be integrated and altered to suit certain requirements, a modular framework can be utilized more often in various settings. We can also examine that our framework is universal, i.e., it is not specific to a particular application domain and facilitates development of a wide range of software projects. Lastly, we can verify that our framework support various business models to generate revenue from its implementation. It can be offered as a B2B, B2C or as a subscription based business model.

VII. CONCLUSION

In this paper, we proposed a novel framework for automated software development using the AI tools. Existing frameworks in the literature lack scalability and modularity, among other constraints. In addition to lacking a viable revenue-generating business model, the majority of them are also limited to a certain application area. In addition, they do not use AI-based code generating tools to keep their costs as low as possible. By capitalizing on the shortcomings of the existing frameworks, our proposed framework simplified and expedited various phases of the software development process through a modular approach. Our proposed framework offered a variety of business models from its adoption to generate profit. We discovered that by using AI technologies, we can reduce the amount of human involvement in the development work, which resulted in minimizing the overall cost related to the software development. Our framework allowed for future improvements and evolution from a scalability perspective, making it simple to add new stages to its life cycle and assist the system in adapting to meet new challenges.

REFERENCES

[1] G. Marzi, "On the nature, origins and outcomes of over featuring in the new product development process," Journal of Engineering and Technology Management, vol. 64, p. 101685, 2022.

[2] M. S. Hamdi, A. Ghannem, and M. Kessentini, "Requirements traceability recovery for the purpose of software reuse: An interactive genetic algorithm approach," Innovations in Systems and Software Engineering, vol. 18, no. 1, pp. 193–213, Mar. 2022.

[3] A. Manjavacas, A. Vizcaíno, F. Ruiz, and M. Piattini, "Global software development governance: Challenges and solutions," Journal of Software: Evolution and Process, vol. 32, no. 10, Apr. 2020.

[4] A. D. Durai, M. Ganesh, R. M. Mathew, and D. K. Anguraj, "A novel approach with an extensive case study and experiment for automatic code generation from the XMI schema of UML Models," The Journal of Supercomputing, vol. 78, no. 6, pp. 7677–7699, Jan. 2022.

[5] D. P. Wangoo, "Artificial Intelligence Techniques in Software Engineering for Automated Software Reuse and Design," 2018 4th International Conference on Computing Communication and Automation (ICCCA), 2018, pp. 1-4.

[6] N. Nguyen and S. Nadi, "An Empirical Evaluation of GitHub Copilot's Code Suggestions," 2022 IEEE/ACM 19th International Conference on Mining Software Repositories (MSR), 2022, pp. 1-5.

[7] A. Dubey, K. Abhinav, S. Jain, V. Arora, and A. Puttaveerana, "HACO: A Framework for Developing Human-AI Teaming," in Proceedings of the 13th Innovations in Software Engineering Conference on Formerly known as India Software Engineering Conference, 2020.

[8] S. Jüngling, M. Peraić, and A. Martin, "Towards AI-based Solutions in the System Development Lifecycle," in AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering, 2020, pp. 1-6.

[9] C. Kaymakci, S. Wenninger, and A. Sauer, "A holistic framework for AI systems in Industrial Applications," Lecture Notes in Information Systems and Organisation, vol. 47, pp. 78–93, 2021.

[10] P. William, P. Kumar, G. S. Chhabra and K. Vengatesan, "Task Allocation in Distributed Agile Software Development using Machine Learning Approach," 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), 2021, pp. 168-172.

[11] H. Wu, Y. Ma, Z. Xiang, C. Yang, and K. He, "A spatial–temporal graph neural network framework for automated software bug triaging," Knowledge-Based Systems, vol. 241, pp. 108308, 2022.

[12] A. Azzazi and M. Shkoukani, "A knowledge-based expert system for supporting security in software engineering projects," International Journal of Advanced Computer Science and Applications, vol. 13, no. 1, pp. 395–400, 2022.

[13] B. Fritscher and Y. Pigneur, "Visualizing Business Model Evolution with the Business Model Canvas: Concept and Tool," 2014 IEEE 16th Conference on Business Informatics, 2014, pp. 151-158.

[14] S. Nastic, A. Morichetta, T. Pusztai, S. Dustdar, X. Ding, D. Vij, and Y. Xiong, "SLOC: Service level objectives for Next Generation Cloud Computing," IEEE Internet Computing, vol. 24, no. 3, pp. 39–50, May 2020. .